PCT

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 17/14	**A1**	(11) International Publication Number: **WO 00/02140** (43) International Publication Date: 13 January 2000 (13.01.00)

(21) International Application Number: PCT/SE99/01224

(22) International Filing Date: 5 July 1999 (05.07.99)

(30) Priority Data:
9802426-8   3 July 1998 (03.07.98)   SE

(71) Applicant: TELEFONAKTIEBOLAGET LM ERICSSON [SE/SE]; S-126 25 Stockholm (SE).

(72) Inventor: SIHLBOM, Björn; Axgatan 24, S-431 40 Mölndal (SE).

(74) Agent: GÖTEBORGS PATENTBYRÅ DAHLS AB; Sjöporten 4, S-417 64 Göteborg (SE).

(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.
With amended claims.

(54) Title: METHOD AND ARRANGEMENT RELATING TO DFT COMPUTATION

(57) Abstract

The present invention relates to an arrangement for a discrete Fourier transform (DFT) computation including m radix-r, $r = 2, 4, 8, ...$, butterfly operators (11.1–11.4), data memory sets (12, 12') comprising memory units (90–93) and switching means (14, 15). The butterfly operators (11.1–11.4, 11.1', 11.2') are arranged in parallel and connected to m memory units ($m = 1, 2, ...$) allowing $2r$ accesses per memory unit during each calculation.

TITLE

METHOD AND ARRANGEMENT RELATING TO DFT COMPUTATION

5

TECHNICAL FIELD OF THE INVENTION

The present invention relates to an arrangement for discrete Fourier transform computation including m radix-r butterfly operators, data memory sets and switching means.

10

The invention also relates to a device including means for discrete Fourier transform computation and method of carrying out the computation.

BACKGROUND OF THE INVENTION

15

For calculation of the Discrete Fourier Transform (DFT), generally the efficient Fast Fourier Transform (FFT) algorithm is used. There are several different methods to calculate FFT and several parameters such as speed, flexibility, complexity etc., affect the way of the implementation.

20

Moreover, there are many different variants of FFT algorithms, for instance from the least complex and least efficient, i.e. radix-2, to more complex and more efficient variant, e.g. radix-4, radix-8, mixed-radix and so on.

25   Normally, the FFT calculation is carried out on one set of data at a time. The length of the data set cannot be arbitrary but can only assume certain values depending on the type of the FFT used.

Generally, the radix 2 and mixed radix variants are most flexible ones since several data lengths
30   are allowed. The drawback with the radix-2 method is its poor efficiency and thereby inferior performance. Mixed-radix suffers from the significant complexity resulting in considerable difficulties when implementing into hardware.

2

The FFT computation has been lately moved into hardware by implementation into special integrated computation circuits.

DESCRIPTION OF RELATED ART

5

FFT computation circuits using different techniques are known through several documents.

US 5,163, 017, for example, discloses a pipelined FFT architecture including a memory for storing complex number data. A pipelined data path is coupled to a memory for accessing
10   complex number data therefrom for computing an FFT butterfly operator and storing the results from the butterfly operator in the memory during one pipeline cycle. The object according to this document is to reduce the number of memory accesses for one butterfly operator. No support for parallel butterfly operators is given.

15   Furthermore, US 5,028,877 discloses a circuit arrangement for the implementation of a fast DFT in real time by using controlled operations of cross-linked butterflies. The arrangement successively transmits two halves of a sequence of a complex input word through a series-parallel input register and an intermediate data storage to a plurality of butterfly operators operating in parallel. The outputs of the operators are switchable by a multiplexer for recursive
20   linkage with the intermediate storage or for delivery of the frequency range-output word to a parallel-series output register. Although parallel butterfly operators are used, according to this document, the solution is not flexible as the FFT length is always determined to 16 points and cannot be changed. The arrangement uses pipeline technique and not memory arrangement for intermediate storage of results. Moreover, the entire sequence of input words subjected to the
25   FFT consists of four times as many as values as there are provided parallel operating butterfly operators. Moreover, no memory access reduction is concerned.

Other arrangement not using parallel butterfly operators and reduced number memory accesses are presented in EP-A1-805 401 and US 4,601,006, the later mainly describing a pipelined
30   architecture for a two-dimensional FFT.

US 4,241,411 describes a device for parallel computation of FFT. M radix-k butterfly operators,

each connected to a corresponding memory unit, operate with an interlaced data set. The device does not provide for an arrangement having variable FFT-length, even though the parallelism is partly known through this device. The butterfly operators are arranged on identical FFT processing cards, which further complicates the use of variable FFT-length. Moreover, the

5   arrangement requires a significant number of memory units for carrying out the parallel computations.

US 4,393,457 discloses an apparatus for generating a specific sequence of addresses of values of an array stored in a digital memory. A first counter which generates a seed value and a

10   second counter which generates a control value generates the addresses. The control value controlling a bit inserting and a programmable shifter to set, respectively, the bit place position of bit insertion and the amount of shift. The output of the bit inserting is the row position of related addresses for butterfly operation of a fast Fourier transform array. The output of the shifter is the address of coefficients associated with the complex rotation of the butterfly

15   operation. No parallel butterfly structure is concerned or suggested. A parallel structure using this apparatus would lead to a very complicated and slow arrangement.

SUMMARY OF THE INVENTION

20   The main object of the present invention is to provide a flexible arrangement for fast computation of the FFT algorithm.

Another main object of the invention is to provide a fast and flexible FFT computation arrangement, in which the FFT length can be determined and changed without a need to modify

25   the arrangement.

Another main object of the invention is to provide a fast and flexible FFT computation arrangement with varying FFT length and a number of parallel operators.

30   The invention also has as one object to provide a FFT computation arrangement, which is suited for hardware implementation, i.e. it is less complex.

4

The arrangement according to the invention provides a new memory configuration, address generation and data control in connection with FFT computations, which allows division of memory sets into smaller units, enabling fewer memory accesses per memory set.

5   Preferably, the arrangement according to the invention can be arranged on a single PCB or even in a single integrated circuit.

Consequently, in the arrangement according to the preamble the butterfly operators are arranged in parallel and m memory units are arranged allowing 2r accesses per memory unit
10   during each calculation. Preferably, to achieve flexibility, the arrangement uses a variable FFT length parameter, wherein the FFT length is r^{LMODE}, where LMODE $\geq m+1$. In one advantageous embodiment using normal bit storing, the number of memory units are 2m and the memory sets are swinging memories.

15   Preferably, the memory size for a memory set is the FFT length divided by the number of the butterfly operators.

In one preferred embodiment the arrangement further includes address generating means and first and second memory control means connected to said memory sets.
20

The address generating means consists of a state-machine, which assumes different states representing selection of different memory configurations in said memory sets, at least one of said states arranges at least one memory set as an input/output memory set and one memory set for receiving data from at least one of said butterfly operators. Preferably, the state-machine is
25   arranged to assume six different states.

To control the addressing of the memories, the first and second memory control means include switching devices controlled by said address generating means, which switching devices comprise multiplexors. The first memory control means, comprising multiplexors are connected
30   between the butterfly operators and memory sets, is arranged to switch data from correct memory sets and the second memory control means is arranged to switch data to correct memory sets. The second memory control means comprises a control signal circuit, an I/O

circuit and switching devices, said control signal circuit and I/O circuit being connected to said switching devices, being controlled by said address generating means. Preferably, the memory sets include four storage means of type SRAM (Static random Access Memory).

5   The invention also relates to a computation device, substantially for discrete Fourier transform (DFT) computation using Fast Fourier Transform (FFT) on a set of data. The device includes m radix-r butterfly operators, data memory sets including memory units and switching means. The device further includes: the butterfly operators arranged in parallel, m memory units, a control block, which controls and supervises functions of the device, a twiddle-coefficient generator,

10   for generating twiddle coefficients to said butterfly operators, memory data control units for controlling data flows to/from the memory sets, means to receive a FFT length for a FFT calculation and a memory controlling unit for controlling the function of the memory units. In one embodiment, the device further includes input data controlling means for processing incoming data, output data controlling means for processing outgoing data, and preprocessing

15   means arranged to process data before the FFT. Preferably, the device may apply both FFT and I FFT (Inverted FFT) on the data. It also includes a processing means performing operations on data in the frequency domain and post-processing means before output. The device is so arranged that the data is read from one memory set and written back to another.

20   The method according to the invention mainly includes the steps of arranging the butterfly operators in parallel and arranging m memory units allowing 2r memory accesses per memory unit during each calculation. According to the method the data is stored in normal or in bit-reversed order in the memory. The FFT calculation consists of a number of calculation stages and that the data flow direction reverses after each stage.

25

Furthermore, the data-flow to the memory sets are controlled by different configurations for determining a pattern to use the switching means in respect of signals STEP, LAP and NFFT = FFT length = r^{LMODE}, LMODE $\geq m+1$, received from a controlling device. The configuration method involves the steps of: determining a first configuration for the first switching means if

30   STEP is zero and LAP is even, determining a second configuration for the first switching means if STEP is zero and LAP is odd, determining a third configuration for the first and the second switching means if STEP is LMODE-2, determining a fourth configuration for the first

6

and the second switching means if STEP is LMODE-1. According to the method the order for
calculation of the butterfly operators is less than 2 data reads/writes from/to 2 port memory
units at each calculation step.

5   Moreover, the data flow to the memory sets is controlled by means of control means, generating
signals for data handling of memories to and from the butterfly operators, the method includes
the steps of: receiving inputs, LAP, which is the calculation step index, STEP which is current
FFT calculation step, checking LAP and if LAP is zero then RLAP initiates a loop index,
generating read commands to said memory sets and generating write commands to said
10   memory sets. If LAP is not zero then other sets of read and write commands are generated. If
data is in bit-reversed order and the read addresses are the same as the write-addresses new
read/write commands are generated.

BRIEF DESCRIPTION OF THE DRAWINGS

15

In the following, the invention will be further described in a non-limiting way under reference
to the accompanying drawings in which:

Fig. 1   is a schematic block diagram of a first embodiment of an arrangement according to
20   the present invention.
Fig. 2   shows another schematic embodiment according to fig. 1 in more detail.
Fig. 3   is a schematic view of DC block according to fig. 2.
Fig. 4   is a schematic view of IDC/ODC blocks according to fig. 3.
Fig. 5   is a schematic view of MDCU-O block according to fig. 2.
25   Fig. 6   is a schematic flow schema over the function of a control block according to fig. 5.
Fig. 7   is a schematic view of MDCU-I block according to fig. 2.
Fig. 8   is a state diagram for a control block MCU according to fig. 2.
Fig. 9   is a schematic view of MU block according to fig. 2.
Fig. 10   illustrates schematically another embodiment according to the invention.
30   DETAILED DESCRIPTION OF THE EMBODIMENTS

Briefly, the present invention uses several parallel butterfly operators for the FFT calculations.

It is possible by division of memory sets (each including a number of memory units) using a new memory configuration, address generation and data flow enabling. By using, for example "swinging " memories, it is for example possible to obtain 8 memory sets with 2 memory accesses during each computation cycle instead of 2 memory sets with 8 accesses. Without

5   swinging memory sets, the number of memory accesses per memory unit will, of coarse, increase.

The main improvements are achieved through a novel data control and memory configuration. At the initial step data is stored in a memory set to be read from. Then a FFT calculation is

10   executed in number of steps. If swinging memories are used, the data change location between the memories by changing the "read" and "write" commands to the memories. During each step the butterfly operators carry out calculations. The serial number of each calculation cycle, the determined FFT-length and the number of the step to be calculated are used as input data for generating memory addresses and data control purpose. Consequently, it is possible to change

15   the FFT-length from calculation to calculation to obtain the best result, for example if the radix-r butterfly operator is used then the possible FFT lengths are r^x, where x is greater or equal to the log2(number of memory units used) +1.

In one non limiting embodiment, which will be described closer in the following, during each

20   calculation cycle 8 data are read from one part of the memory set while 8 data are written to the other part of the memory set. In this case 4 memory units having 2 data inputs (memory with 2 ports [inputs/outputs]) are used. The memory size for a memory set having 2 data inputs is the FFT length divided by the number of the butterfly operators. The basis for the calculation order of the butterfly operators is to avoid more than 2 data read/write from/to each 2 port memory

25   each calculation step.

The data read from one side of the memory sets appear in a special order depending on the calculation step, FFT length and the type of the calculations that must be carried out. This is achieved by means of switching arrangements, which switch data to a correct butterfly operator

30   and then back to the order the data is written to the memory.

The arrangement according to the present invention, hereinafter called P3FTD (Parallel

8

Flexible Fast Fourier Transform Device) 10 is illustrated schematically in the fig. 1. P3FTD includes Butterfly Operators (BOs) 11.1-11.4, memory sets (MSs) 12, address generating device MCU 13, input data controller 14 (IDC), output data controller 15 (ODC), a twiddle coefficient generator 16 and data flow controllers 17 and 18. P3FTD 10 may further include

5   input and output buffers 28 and 29, respectively, for example in form of FIFO (First In First Out) memory units. However, the novel core of the invention comprises the memory blocks 12, address generating device MCU 13 and input/output data controllers 14 and 15 and parallel butterfly operators.

10   A preferred embodiment of P3FTD as it may be implemented in an integrated circuit is shown in more detail in the block diagram of fig. 2. For clarity reason, only some internal signals are shown. Arrows not filled denote control signals.

The used architecture is based on four (4) radix-2 BOs, which are located in the Data

15   Computation (DC) block 11. Data is read from one memory set (MS0, MS1, MS2) 12.1-12.3 and written back to another (MS0, MS1, MS2) after a butterfly operation. The FFT calculation consists of a number of calculation stages. The data direction reverses after each stage. One of the MSs is used for storing data, substantially for simultaneous input/output.

20   The four BOs are provided with twiddle-coefficients from TG 16. When data is read from MS, it does not appear in a correct order for storing back in the MS after the butterfly operation. The Memory Data Control Units MDCU-I 17 (input) and MDCU-O 18 (output) control the data flow to/from the memory sets. The MSs "switch places", i.e. the data is not read and written to same MSs, which is handled by MDCU-I 17, MDCU-O 18. The function of these circuits is

25   further controlled by a memory controlling unit (MCU) 13. An additional memory set, ICM, 20 is used to hold some control parameters, such as filter and window coefficients used for butterfly operations.

The embodiment is provided by several other control blocks: the incoming data to be processed

30   is first handled by IOLC 21, which may be part of the in buffer 28 and the output data is handled by the block DOC 22, which may be part of the output buffer 29. Moreover, the block PRE 23 is arranged to process data before the FFT in DC 11 block.

9

In one preferred embodiment it is possible to apply both FFT and I FFT (Inverted FFT) on the data. In this case a block FDO 24 performs operations on data in the frequency domain. The results are processed by a post-processing unit POST 25 before output.

5   The CONTROL block 26 is an internal or external controlling unit, which controls and supervises substantially all functions of P3FTD. If it is provided internally, it also may handle communication with the external controller(s). Yet another control block, PROC 27 may be arranged, which is a supervising block controlling the functions of DC 11, MDCU-I 17, MDCU-O 18, TG 16 and possible FFT/IFFT functions.

10

In the following the different blocks illustrated in fig. 2 will be described more closely.

CONTROL 26

Preferably, the CONTROL 26 consists of a state-machine, substantially controlling all

15   functions of P3FTD and a possible communication section, which handles the communication with external controllers. The control signals from this block will be described in conjunction with the description of remaining blocks.

The CONTROL block also includes a register into which the FFT lengths used for the

20   calculations are transferred. The FFT length is an application depending variable and can be obtained from external control arrangements, for example a radar receiver processing unit or a video signal processing unit.

IOLC 21

25   The IOLC block 21 is arranged to read data from the input, which can be a buffer or register (preferably FIFOs) and write it to the data memory MS, which at that time is configured as I/O memory by MCU 13. IOLC 21 is implemented as a state-machine controlling the external FIFOs (not shown) connected to the input port. It produces data to the PRE 23. It is initiated by CONTROL 26. Data is received in batches. Preferably, the "Read" operation from the FIFOs is

30   possible at the clock-frequency rate.

DOC 22

The DOC block 22 (Data Output Controller) is arranged to transmit data to the output buffers, which can be implemented as a FIFO. Preferably, the block contains a synchronous internal FIFO for reading data (part of data or a data header). A control unit, preferably implemented as

5   a state-machine, reads data from the internal data memories and writes to the external output (FIFO). The block may also include a device for converting data to different forms (parallel/serial) or protocols receivable by external units.

Output data is received from POST 25 and written into a FIFO (not shown). The DOC 22 reads

10   data by sending addresses (data location) to MDCU-O 18. The output address to MDCU-O is updated and communication between MDCU-O and ODC is enabled by an update signal. To receive data from MDCU-O 18 to DOC 22 an initiation signal is activated and DOC is updated each cycle until a data valid signal from POST 25 is received. At least parts of data (first bytes) are written to an output FIFO (not shown). DOC is then updated and enabling signals are

15   initiated before the next set of data is valid on the input. This procedure is repeated until the last set has been written to the OFIFO. When the last address is sent to MDCU-O, the address valid signals are disabled. The last set of data is detected when a data valid signal from POST 25 is disabled. If OFIFO signals full or a hold signal is active, the writing to OFIFO is paused.

20   *PRE 23*

As mentioned above, the PRE block 23 (PRE-processing) performs operations on data before the FFT/IFFT operations. It calculates complex values including multiplication operations etc.

FDO 24

25   The FDO block 24 (Frequency Domain Operator) performs operations on data in the frequency domain between the FFT/IFFT operations.

ICM 20

The ICM block 20 is a memory set, which holds parameters for filtering, window coefficients

30   etc. An external processor or internal units are able to access this memory for retrieving or changing the parameters. The ICM has the following operation modes:

–   an External write mode,

–   a FDO read mode, and

–   a PRE read mode.

In a preferred embodiment the ICM consists of a 4096x32 synchronous SRAM, with a

5   bidirectional common input and output data bus. The write-signal controls the direction of the
data bus.

POST 25

The POST block 25 executes the final operations on the data before the result is delivered from

10   P3FTD. POST 25 includes, for instance means for performing functions, such as scaling,
rounding and clipping.

DC 11

The DC block 11 is the main signal processing block of the P3FTD and in this embodiment,

15   shown in detail in fig. 3, consist of 4 radix-2 butterfly operators (BOs)11.1-11.4, input data
flow controller IDC 14, output data flow controllers ODC 15, control unit (CU) 30 and delay
elements 31 and 32.

The delay elements are arranged to compensate control signals for DC 11 delays. IDC 14 and

20   ODC 15 perform the data switching and the CU 30, among others controls one stage
calculations.

The architecture of IDC 14 and ODC 15 is illustrated in fig. 4. Both blocks are identical and
consist of several multiplexors 40-47, where first and last MUXs 40 and 47, respectively, in this

25   configuration, have two data inputs while the remaining MUXs have four inputs.

The input data to IDC 14 originates from MDCU-O 18. There are 8 input signals $IDC_0,..,IDC_7$
including complex signals (real and imaginary). The outputs from IDC 14 are inputs to BOs
11.1 to 11.4 and denoted BO_nI_i, where n is the BO number and i (i=0 or 1) is the signal number.

30   BO_1I_1 means input data to BO 1 (11.1) input 1. Inputs to ODC 15 are outputs from the BOs
11.1 to 11.4, hence, in analogy with above BO_nO_i means output from BO number n
(11.n=1,..,4) and signal number i (i= 0 or 1).

Input signals to the inputs of the MUXs (MUX0-MUX7 corresponding to MUXs 40-47) are

listed in Table 1.

Table 1

	MUX0	MUX1	MUX2	MUX3	MUX4	MUX5	MUX6	MUX7
0	IDC_0 BO_0O_0	IDC_1 BO_0O_1	IDC_0 BO_0O_0	IDC_3 BO_1O_1	IDC_1 BO_0O_1	IDC_3 BO_1O_1	IDC_1 BO_0O_1	IDC_5 BO_2O_1
1	IDC_2 BO_0O_0	IDC_2 BO_1O_0	IDC_1 BO_0O_1	IDC_4 BO_2O_0	IDC_2 BO_1O_0	IDC_5 BO_2O_1	IDC_3 BO_1O_1	IDC_7 BO_3O_1
10	✕	IDC_4 BO_2O_0	IDC_2 BO_1O_0	IDC_5 BO_2O_1	IDC_3 BO_1O_1	IDC_6 BO_3O_0	IDC_5 BO_2O_1	✕
11	✕	IDC_6 BO_3O_0	IDC_4 BO_2O_0	IDC_6 BO_3O_0	IDC_4 BO_2O_0	IDC_7 BO_3O_1	IDC_6 BO_3O_0	✕

Outputs from the MUXs to next block are listed in Table 2.

Table 2

	MUX0	MUX1	MUX2	MUX3	MUX4	MUX5	MUX6	MUX7
ODC	ODC_0	ODC_1	ODC_2	ODC_3	ODC_4	ODC_5	ODC_6	ODC_7
BO	BO_0I_0	BO_0I_1	BO_1I_0	BO_1I_1	BO_2I_0	BO_2I_1	BO_1I_0	BO_1I_1

Control signals to IDC 14 and ODC 15 are input data index from MDCU-I (delayed to ODC) and control signals from CONTROL and PROC blocks.

The output of each MUX is determined according to table 3. In table 3, the columns 2-8 show the MUXs' inputs, selected based on control words (configurations) 0-5 (the first column from left). There are 6 sets of control patterns for the MUXs.

Table 3

#	Mux7	Mux6	Mux5	Mux4	Mux3	Mux2	Mux1	Mux0
0	1	1	1	0	11	10	10	0
1	0	0	11	10	1	0	11	1
2	1	10	10	11	0	1	1	0
3	1	11	1	11	0	10	0	0
4	1	1	10	1	10	1	10	0
5	1	10	0	0	11	11	1	0

From Table 3, it appears that control signals to Mux4 - Mux7 are inverted and reversed signals of the Mux0 - Mux3, which fact simplifies the control of the MUXs further.

The control words 0-5 (the first column of table 3) determining the pattern to be used in IDC and ODC in respect of the input signals STEP and LAP are set forward in table 4:

Table 4

Control signal	Control word	
	IDC	ODC
STEP=0, LAP(0)=0	0	3
STEP=0, LAP(0)=1	1	3
STEP=LMODE-2	2	2
STEP=LMODE-1	4	5
else	3	3

STEP is the current FFT calculation step between 0 and log2(NFFT)-1, LAP is the calculation step index (calculation cycle counter) between 0 and NLAP-1 (number of laps), NFFT is the FFT length, in this example 8 up to 4096, NLAP=NFFT/8 and LMODE is FFT length mode obtained from CONTROL. A calculation cycle is a one butterfly calculation executed by each BO.

From table 4 follows that:

if STEP is 0 (zero) and LAP is even then the configuration 0 is determined for IDC,

if STEP is 0 (zero) and LAP is odd then configuration 1 is determined IDC,

if STEP is LMODE-2 then configuration 2 is determined for IDC and ODC,

if STEP is LMODE-1 then configuration 4 is determined for IDC and configuration 5 for ODC, otherwise configuration 3 is determined for both IDC and ODC.

BO 11

Back to fig. 3, each BO is a radix-2 butterfly unit. A radix-2 butterfly takes two complex inputs and a complex twiddle coefficient (W0 - W3) and produces two complex outputs. The twiddle coefficient depends on the butterfly position in the calculation scheme. The function of the butterfly operator is assumed to be known to a person skilled in the art and not described closer

here.

TG 16

The TG block 16 calculates the butterfly twiddle coefficients for each of the four parallel
butterfly-units. TG 16 includes a circuit for generating index values, which depend on which
butterfly is to calculate the FFT calculation scheme. Additionally, TG includes a circuit for
generating the twiddle-coefficients W_0, W_1, W_2 and W_3. The BOs that are to be calculated each
stage and LAP are given by 4 index values calculated according to the following method. The
four complex twiddle coefficients are then calculated as:

$$W(n) = EXP (j * 2 * \pi * D * INDEX(n) / 2^{(1+STEP)}),$$

where j is the imaginary part of the data and D is the FFT direction, and

$$INDEX(n)=n*NLAP + LAP + \delta(STEP) * (n \text{ MODULUS } 2) * (NLAP-2*LAP-1)$$

δ is Kroneckers delta function, defined as:

$$\delta(x)= 0, x \neq 0; \text{ and } \delta(x) = 1, x = 0.$$

STEP, NLAP and LAP are defined above.

CU 30

The CU 30 is a state machine producing index values to IMC 20, which calculates the addresses
to memory sets. Data is passed through the ODC and arrives at DC, then it is calculated and fed
to IDC to be stored in another MS.

MS 12

A memory set, MS, 12 is closely shown in fig. 9. At least 3 MSs 12.1-12.3 are arranged, one
operating as the I/O memory. The MS 12 in this embodiment consist of 4 memory units, here 4
dual-port SRAMs (Static Random Access Memory) 90-93.

The input data to DIN0 and DIN1 port of each SRAM 90-93 is delivered from data outputs of
MDCU-O 18 and outputs of each SRAM, DOUT0, DOUT1, are connected to data inputs of
MDCU-I 17. The control signals (address and write signals A0, WE0, A1, WE1) are supplied
from MDCU-O 18.

MDCU-O 18

The main function of the MDCU-O 18 is to switch data and addresses to correct memory sets

and memory locations. A somewhat detailed embodiment of the MDCU-O is shown in fig. 5. Generally, MDCU-O consists of a control signal circuit (RW) 50, an I/O circuit (IO) 51 and MUXs 52-54.

5   IO 51 generates I/O address vectors and its outputs are inputs to the in-ports of the MUXs 52-54 (port 2). The inputs to IO 51 originate from DOC 22 and PRE 23, for instance including read and data valid signals.

      The remaining in-ports (0, 1) of the MUXs are connected to the outputs of the RW 50 and
10   provided with read/write signals to be transmitted to MSs. MUX select signals to each MUX originates from MCU 13. The outputs of the MUXs 52-54 are connected to MSs 12.1-12.3, respectively, for switching address vectors, data, read address and address indexes.

      RW 50 generates signals for data handling of memories to and from DC 11. The input signals
15   to this circuit include data and data control signals from DC 11 and FDO 24. The control signals outputted to MUXs 52-54 for controlling MS reads/writes are generated according to the flow diagram shown in fig. 6.

      The procedure has the following inputs: NFFT the FFT length, e.g. 8 to 4096, number of FFT,
20   LAP, which is the calculation step index between 0 and NLAP-1, and STEP, which is the STEP index between 0 and log2(NFFT)-1.

      There are also arranged help variables, 62, N=log2(NFFT)-3, S=6-N and NLAP=NFFT/8, 61

25   During the calculation of one FFT STEP, NFFT/2 butterfly operations perform each STEP. There are 4 BOs. Therefore the number of laps (NLAP) is NFFT/2/4.

      The procedure returns outputs:
              W00, W01, W10, W11, write addresses, and
30   R00, R01, R10, R11, read addresses.

      The first index denotes address for port 0 or 1 on SRAMs 90-93, while the second index denotes whether the address is intended for SRAM 90 and SRAM 92 (index=0) or for SPRAM 91 and SRAM 93 (index=1). W01 is, for example the write address to port 0 on SRAM 91 and

SRAM 93, R10 is the read address to port 1 on SRAM 90 and SRAM 92.

According to the procedure if the data is in bit-reversed order in the memory, the calculations can be done 'in-place' and the 'swinging memory' structure is not required. The address generation is given below in two versions, with and without having data in bit-reversed order.

In normal order the input data is put in the memory, i.e. first data in index 0 and second at index 1 etc. This results in that for the first stage, data must be read in bit-reversed order and written into normal order. In this case swinging memories must be used, since the calculation of the first stage isn't performed 'in-place'. It is significant to read data in bit-reversed order, since data is divided into four memory units each containing one quarter of the input data set. It is also necessary to calculate the butterfly operations in an order in respect of that only two data are read and written to/from each memory unit. Therefore, the first stage is processed differently from the subsequent stages. RLAP is an intermediate index that is used for read address calculations for the first stage.

Then the procedure 60 continues by checking LAP and if LAP is zero, 63, then RLAP is initiated, 64:

$RLAP(N-1 \text{ down to } 1) = LAP(1 \text{ up to } N-1) \text{ xor } LAP(0)$

$RLAP(0) = LAP(0)$

then read commands are generated, 65,

$R00 = 2*RLAP$

$R10 = R00+1$

$R01$ = bitwise inversion of $R10(N \text{ down to } 0)$, remaining bits of $R01$ is '0'

$R11$ = bitwise inversion of $R00(N \text{ down to } 0)$, remaining bits of $R11$ is '0'

then write commands are generated, 66,

$W00 = 2*LAP$

$W10 = W00+2^{\text{minimum}(STEP,N)}$

$W01$ = bitwise inversion of $W10(N \text{ down to } 0)$, remaining bits of $W01$ is '0'

$W11$ = bitwise inversion of $W00(N \text{ down to } 0)$, remaining bits of $W11$ is '0'

If LAP is not zero, 63, i.e. not first time then read commands are generated, 67

$R00 = 2*LAP-LAP(N \text{ down to } 0)$

$R10 = R00+2^{\text{minimum}(STEP, N)}$

$$R01 = R00$$

$$R11 = R10$$

and finally write commands are generated, 68,

$$W00 = 2*LAP\text{-}LAP(N \text{ downto } 0)$$

$$W10 = W00 + 2^{minimum(STEP, N)}$$

$$W01 = W00$$

$$W11 = W10$$

The procedure is terminated by outputting the generated commands, 69.

In case of Bit-reversed order, the data can be processed in-place, since it is already placed in bit-reversed order in the memory. The read addresses are the same as the write addresses in the example above. The fist stage is also calculated:

$$R00 = W00 = 2*LAP\text{-}LAP(N \text{ down to } 0)$$

$$R10 = W10 = W00 + 2^{minimum(STEP, N)}$$

$$R01 = W01 = W00$$

$$R11 = W11 = W10.$$

If this method is used only the second branch will be used in TG 16.

MDCU-I 17

The MDCU-I 17 is provided to switch data from the correct memory-unit 12 and memory location. It is illustrated in fig. 7. It also selects data from the memory sets to POST 25, FDO 24 and DC 11. The block includes several switching units, preferably MUXs 70-76. The control input signals to the MUXs are provided by MCU13.

MUXs 70 and 75 switch data index signals from MSs 12.1-12.3, MUX 71 switches data from MSs and MUXs 72 and 76 switch data valid signals from the MSs. MCU 13 controls the MUXs' switching operations. The outputs from the MUXs are further switched to FDO or POST through MUXs 73 and 74.

MCU 13

The MCU block 13 consists of a state-machine. The memory configuration is arranged to assume six different states, S0-S5 as shown in fig. 8.

In different states, different MSs are selected. The selections are disclosed in Table 5. MSs 12.1-12.3 are indicated with MS0-MS2, respectively.

Table 5

STATE	I/O MEMORY	DATA FROM MEMORY TO BOs	DATA FROM BOs TO MEMORY
S0	MS0	MS2	MS1
S1	MS0	MS1	MS2
S2	MS1	MS0	MS2
S3	MS1	MS2	MS0
S4	MS2	MS1	MS0
S5	MS2	MS0	MS1

Corresponding multiplexor-port select-signals are disclosed in Table 6, the MUX-select-signals are indicated in figs. 5 and 7. According to the table 6, at state S0, MS0 is used as I/O memory, content of MS2 is sent to BOs and results from BOs are written to MS1.

Table 6

SELECT STATE	MS0_SEL (fig. 5)	MS1_SEL (fig. 5)	MS2_SEL (fig. 5)	DP_SEL (fig. 7)	O_SEL (fig. 7)
S0	2	1	0	2	0
S1	2	0	1	1	0
S2	0	2	1	0	1
S3	1	2	0	2	1
S4	1	0	2	1	2
S5	0	1	2	0	2

The values in the cells of the table 5 refer to the port number of a MUX to be selected. Accordingly, S2 and MS1_SEL means that port 2 of MUX 53 (fig. 5) is selected.

In the state diagram of fig. 8 two state-alteration signals are indicated, MEM_SWITCH and NEW_STATE. MEM_SWITCH is a control signal from CONTROL 26 to switch all memories

and NEW_STATE is a control signal from PROC to switch read and write memories. The initial state, S0 is reached after a reset command, REST_CMD.

The embodiment according to fig. 2 and relating drawings is only given as a non limiting
5   example. It is possible to insert new or cancel some blocks/circuits and arrange some blocks as external circuits. The multiplexors can be substituted by other switching arrangements, such as PLC (Programmable Logic Control) or the like.

Another embodiment is shown in fig. 10, in which two radix-4 butterfly operators 11.1' and
10   11.2' are provided. The number of memory sets 12' is reduced to four but the number of data inputs of each memory set is increased to four. Same reference numbers denote same parts as in fig.1. However, reference number of parts having modified functions due to radix-4 butterfly operators are denoted with accents otherwise the function of all parts is analogous to the above described embodiment.

15

The invention is not limited the shown embodiments but can be varied in a number of ways without departing from the scope of the appended claims and the arrangement and the method can be implemented in various ways depending on application, functional units, needs and requirements etc.

20

CLAIMS

1. An arrangement for a discrete Fourier transform (DFT) computation including m radix-r, r = 2, 4, 8, ..., butterfly operators (11.1-11.4), data memory sets (12, 12') comprising memory units (90-93) and switching means (14, 15),

characterised in,

that said butterfly operators (11.1-11.4, 11.1', 11.2') are arranged in parallel and connected to m memory units (m = 1,2,...) allowing 2r accesses per memory unit during each calculation step, and that it uses a variable FFT length parameter for butterfly operations.

2. An arrangement according to claim 1,

characterised in,

that the FFT length is r^{LMODE}, wherein LMODE $\geq m+1$.

3. An arrangement according to claim 1,

characterised in,

that the number of memory units are 2m.

4. An arrangement according to claim 1,

characterised in,

that a memory size for a memory set is the FFT length divided by the number of the butterfly operators.

5. An arrangement according to claims 3,

characterised in,

that said memory sets (12, 12') are swinging memories.

6. An arrangement according to any of claims 1-5,

characterised in,

that it includes address generating means, and first and second memory control means (17, 18) connected to said memory sets (12,12').

7. An arrangement according to claim 6,

characterised in,

that said address generating means (13) consists of a state-machine.

8. An arrangement according to claim 7,

characterised in,

5   that said state-machine assumes different states (S0-S5) representing selection of different

memory configurations in said memory sets (12).

9. An arrangement according to claim 8,

characterised in,

10   that at least one of said states arranges at least one memory set as an input/output memory set

and one memory set for receiving data from at least one of said butterfly operators.

10. An arrangement according to claim 8,

characterised in,

15   that said state-machine is arranged to assume six different states (S0-S5).

11. An arrangement according to claim 6,

characterised in,

that said first and second memory control means include switching devices controlled by said

20   address generating means.

12. An arrangement according to claim 11,

characterised in,

that said switching devices comprise multiplexors (40-43).

25

13. An arrangement according to claim 6,

characterised in,

that said first memory control means (17) is arranged to switch the data from appropriate

memory sets (12, 12').

30

14. An arrangement according to claim 6,

characterised in,

that said second memory control means (18) is arranged to switch data to appropriate memory

sets (12, 12').

15. An arrangement according to claim 6,

characterised in,

that said first memory control means (17) comprises multiplexor (70-76) connected between the
butterfly operators and memory sets.

5

16. An arrangement according to claim 6 and 8,

characterised in,

that said second memory control means (18) comprises a control signal circuit (50), an I/O
circuit (51) and switching devices (52-54), said control signal circuit and I/O circuit being

10   connected to said switching devices (52-54), being controlled by said address generating
means.

17. An arrangement according to claim 1,

characterised in,

15   that the memory sets (12) include four storage means of type SRAM (Static random Access
Memory)(90-91).

18. An arrangement according to claim 1 and 6,

characterised in,

20   that said switching means (14, 15) include multiplexors (40-47) arranged between the butterfly
operators and the memory controlling means (17, 18).

19. An arrangement according to claim 18,

characterised in,

25   that output of said switching means (14, 15) depends on current FFT calculation step and a
calculation step index.

20. A computation device, substantially for discrete Fourier transform (DFT) computation
using Fast Fourier Transform (FFT) on a set of data, the device including m radix-r butterfly

30   operators (11.1-11.4), data memory sets (12) including memory units and switching means (14,
15),

characterised in,

that said butterfly operators are arranged in parallel and the device further includes:

— m memory units, $m=0,1,2,...,$

- a control block (26, 27), which controls and supervises functions of the device,
- a twiddle-coefficient generator (16), for generating twiddle coefficients (W0-W3) to said butterfly operators,
- Memory Data Control Units (17, 18) for controlling data flows to/from the memory sets,
- means (26) to receive a FFT length for a FFT calculation, and
- a memory controlling unit (13) for controlling the function of the memory units.

21. The device according to claim 20,

characterised in,

that said FFT length is variable: $FFT = r^{LMODE}$, wherein $LMODE \geq m+1$.

22. The device according to claim 20,

characterised in,

that it further includes

- input data controlling means (21) for processing incoming data,
- output data controlling means (22) for processing outgoing data, and
- preprocessing means (23) arranged to process data before the FFT.

23. The device according to claim 20,

characterised in,

that it applies both FFT and I FFT (Inverted FFT) on the data.

24. The device according to claim 22,

characterised in,

that it includes a processing means (24) performing operations on data in the frequency domain.

25. The device according to claim 23,

characterised in,

that it includes post-processing means (25) before output.

26. The device according to claim 23,

characterised in,

that data is read from one memory set (12.1-12.3) and written back to another.

27. A method for carrying out a FFT calculation in a computation arrangement including m
(m=1,2,3,...) radix-r butterfly operators (11.1-11.4, 11.1', 11.2'), data memory sets (12, 12')
comprising memory units, and first and second switching means (14, 15),
characterised by,

5   arranging said butterfly operators (11.1-11.4, 11.1', 11.2') in parallel and arranging m memory
units (12) allowing 2r memory accesses per memory unit during each calculation.

28. A method according to claim 27,
characterised in,

10   that data is stored in normal or in bit-reversed order in the memory.

29. A method according to claim 27,
characterised in,

that the FFT calculation consists of a number of calculation stages and that the data flow
15   direction reverses after each stage.

30. A method according to claim 29,
characterised in,

that data-flow to the memory sets are controlled by different configurations for determining a
20   pattern to use the switching means (14, 15) in respect of signals STEP, LAP and NFFT = FFT
length = r^{LMODE}, LMODE ≥m+1, received from a controlling device (26, 27).

31. A method according to claim 30,
characterised by,

25   --   determining a first configuration for the first switching means (14) if STEP is 0
                (zero) and LAP is even,
        --   determining a second configuration for the first switching means (14) if STEP is 0
                (zero) and LAP is odd,
        --   determining a third configuration for the first and the second switching means (14;
30   15) if STEP is LMODE-2,
        --   determining a fourth configuration for the first and the second switching means (14;
                15) if STEP is LMODE-1.

32. A method according to claim 27,

characterised in,

that the data flow to the memory sets is controlled by means of control means, generating

signals for data handling of memories to and from the butterfly operators, the method including

the steps of:

5   —   receiving inputs: LAP, which is the calculation step index, STEP which is current

                  FFT calculation step

    —   checking LAP and if LAP is zero (63) then RLAP initiating a loop index,

                  wherein RLAP(N-1 down to 1) = LAP(1 up to N-1) xor LAP(0), and

                  RLAP(0) = LAP(0),

10   —   generating read commands to said memory sets,

                       R00 = 2*RLAP,

                       R10 = R00+1,

                       R01 = bitwise inversion of R10(N down to 0), remaining bits of R01 is

                       '0',

15   R11 = bitwise inversion of R00(N down to 0), remaining bits of R11 is

                       '0',

    —   and generating write commands to said memory sets,

                       W00 = 2*LAP,

                       W10 = W00+2 $^{minimum(STEP,N)}$,

20   W01 = bitwise inversion of W10(N down to 0), remaining bits of W01

                       is '0',

                       W11 = bitwise inversion of W00(N down to 0), remaining bits of W11

                       is '0',

          where the first index denotes address for ports of a memory unit (90- 93) and the

25   second index denotes address to a memory units (90-93)

33. A method according to claim 32,

characterised in,

that if LAP is not zero then read commands are generated:

30   R00 = 2*LAP-LAP(N down to 0),

                R10 = R00+2 $^{minimum(STEP,N)}$,

                R01 = R00,

                R11 = R10,

and generating write commands:

$$W00 = 2*LAP\text{-}LAP(N \text{ downto } 0),$$

$$W10 = W00 + 2^{minimum(STEP,N)},$$

$$W01 = W00,$$

$$W11 = W10.$$

5

34. A method according to claim 28 and 30,

characterised in,

that data is in bit-reversed order and the read addresses are the same as the write addresses:

$$R00 = W00 = 2*LAP\text{-}LAP(N \text{ down to } 0),$$

10

$$R10 = W10 = W00 + 2^{minimum(STEP,N)},$$

$$R01 = W01 = W00,$$

$$R11 = W11 = W10.$$

35. Method according to claim 27,

15   *characterised in,*

that an order for calculation of the butterfly operators is less than 2 data reads/writes from/to 2 port memory units at each calculation step.

AMENDED CLAIMS

[received by the International Bureau on 03 December 1999 (03.12.99);
original claims 1-35 replaced by amended claims 1-33 (7 pages)]

1. An arrangement for a discrete Fourier transform (DFT) computation including m radix-r, r = 2, 4, 8, ..., butterfly operators (11.1-11.4), data memory sets (12, 12') comprising memory units (90-93) and switching means (14, 15),

characterised in,

that said butterfly operators (11.1-11.4, 11.1', 11.2') are arranged in parallel and connected to m memory units (m = 1,2,...) allowing 2r accesses per memory unit during each calculation step, that it uses a variable FFT length parameter for butterfly operations wherein the FFT length is r^{LMODE}, wherein LMODE $\geq$ m+1.

2. An arrangement according to claim 1,

characterised in,

that the number of memory units are 2m.

3. An arrangement according to claim 1,

characterised in,

that a memory size for a memory set is the FFT length divided by the number of the butterfly operators.

4. An arrangement according to claims 2,

characterised in,

that said memory sets (12, 12') are swinging memories.

5. An arrangement according to any of claims 1-4,

characterised in,

that it includes address generating means, and first and second memory control means (17, 18) connected to said memory sets (12,12').

6. An arrangement according to claim 5,

characterised in,

that said address generating means (13) consists of a state-machine.

AMENDED SHEET (ARTICLE 19)

28

7. An arrangement according to claim 6,

characterised in,

that said state-machine assumes different states (S0-S5) representing selection of different

memory configurations in said memory sets (12).

5

8. An arrangement according to claim 7,

characterised in,

that at least one of said states arranges at least one memory set as an input/output memory set and

one memory set for receiving data from at least one of said butterfly operators.

10

9. An arrangement according to claim 7,

characterised in,

that said state-machine is arranged to assume six different states (S0-S5).

15   10. An arrangement according to claim 5,

characterised in,

that said first and second memory control means include switching devices controlled by said

address generating means.

20   11. An arrangement according to claim 10,

characterised in,

that said switching devices comprise multiplexors (40-43).

12. An arrangement according to claim 5,

25   *characterised in,*

that said first memory control means (17) is arranged to switch the data from appropriate memory

sets (12, 12').

13. An arrangement according to claim 5,

30   *characterised in,*

that said second memory control means (18) is arranged to switch data to appropriate memory sets

(12, 12').

AMENDED SHEET (ARTICLE 19)

14. An arrangement according to claim 5,

characterised in,

that said first memory control means (17) comprises multiplexor (70-76) connected between the butterfly operators and memory sets.

5

15. An arrangement according to claim 5 and 7,

characterised in,

that said second memory control means (18) comprises a control signal circuit (50), an I/O circuit (51) and switching devices (52-54), said control signal circuit and I/O circuit being connected to

10   said switching devices (52-54), being controlled by said address generating means.

16. An arrangement according to claim 1,

characterised in,

that the memory sets (12) include four storage means of type SRAM (Static random Access

15   Memory)(90-91).

17. An arrangement according to claim 1 and 5,

characterised in,

that said switching means (14, 15) include multiplexors (40-47) arranged between the butterfly

20   operators and the memory controlling means (17, 18).

18. An arrangement according to claim 17,

characterised in,

that output of said switching means (14, 15) depends on current FFT calculation step and a

25   calculation step index.

19. A computation device, substantially for discrete Fourier transform (DFT) computation using Fast Fourier Transform (FFT) on a set of data, the device including m radix-r butterfly operators (11.1-11.4), data memory sets (12) including memory units and switching means (14, 15),

30   *characterised in,*

that said butterfly operators are arranged in parallel and the device further includes:

− m memory units, $m=0,1,2,...,$

−   a control block (26, 27), which controls and supervises functions of the device,

AMENDED SHEET (ARTICLE 19)

– a twiddle-coefficient generator (16), for generating twiddle coefficients (W0-W3) to said butterfly operators,

– Memory Data Control Units (17, 18) for controlling data flows to/from the memory sets,

– means (26) to receive a variable FFT length for a FFT calculation, said variable FFT length being FFT = r^{LMODE}, wherein LMODE $\geq$ m+1., and

– a memory controlling unit (13) for controlling the function of the memory units.

20. The device according to claim 19,

characterised in,

that it further includes

– input data controlling means (21) for processing incoming data,

– output data controlling means (22) for processing outgoing data, and

– preprocessing means (23) arranged to process data before the FFT.

21. The device according to claim 19,

characterised in,

that it applies both FFT and I FFT (Inverted FFT) on the data.

22. The device according to claim 20,

characterised in,

that it includes a processing means (24) performing operations on data in the frequency domain.

23. The device according to claim 21,

characterised in,

that it includes post-processing means (25) before output.

24. The device according to claim 21,

characterised in,

that data is read from one memory set (12.1-12.3) and written back to another.

25. A method for carrying out a FFT calculation in a computation arrangement including m (m=1,2,3,...) radix-r butterfly operators (11.1-11.4, 11.1', 11.2'), data memory sets (12, 12') comprising memory units, and first and second switching means (14, 15),

AMENDED SHEET (ARTICLE 19)

characterised by,

arranging said butterfly operators (11.1-11.4, 11.1', 11.2') in parallel, arranging m memory units (12) allowing 2r memory accesses per memory unit during each calculation, and arranging means (26) to receive a variable FFT length for a FFT calculation, said variable FFT length being

FFT = r^{LMODE}, wherein LMODE $\geq$ m+1..

26. A method according to claim 25,

characterised in,

that data is stored in normal or in bit-reversed order in the memory.

27. A method according to claim 25,

characterised in,

that the FFT calculation consists of a number of calculation stages and that the data flow direction reverses after each stage.

28. A method according to claim 27,

characterised in,

that data-flow to the memory sets are controlled by different configurations for determining a pattern to use the switching means (14, 15) in respect of signals STEP, LAP and NFFT = FFT length = r^{LMODE}, LMODE $\geq$ m+1, received from a controlling device (26, 27).

29. A method according to claim 28,

characterised by,

–   determining a first configuration for the first switching means (14) if STEP is 0 (zero) and LAP is even,

–   determining a second configuration for the first switching means (14) if STEP is 0 (zero) and LAP is odd,

–   determining a third configuration for the first and the second switching means (14; 15) if STEP is LMODE-2,

–   determining a fourth configuration for the first and the second switching means (14; 15) if STEP is LMODE-1.

30. A method according to claim 25,

AMENDED SHEET (ARTICLE 19)

characterised in,

that the data flow to the memory sets is controlled by means of control means, generating signals for data handling of memories to and from the butterfly operators, the method including the steps of:

5   −   receiving inputs: LAP, which is the calculation step index, STEP which is current FFT calculation step

    −   checking LAP and if LAP is zero (63) then RLAP initiating a loop index, wherein RLAP(N-1 down to 1) = LAP(1 up to N-1) xor LAP(0), and RLAP(0) = LAP(0),

10   −   generating read commands to said memory sets,

        $R00 = 2*RLAP$,

        $R10 = R00+1$,

        $R01$ = bitwise inversion of R10(N down to 0), remaining bits of R01 is '0',

        $R11$ = bitwise inversion of R00(N down to 0), remaining bits of R11 is '0',

15   −   and generating write commands to said memory sets,

        $W00 = 2*LAP$,

        $W10 = W00+2^{minimum(STEP,N)}$,

        $W01$ = bitwise inversion of W10(N down to 0), remaining bits of W01 is '0',

        $W11$ = bitwise inversion of W00(N down to 0), remaining bits of W11 is '0',

20   where the first index denotes address for ports of a memory unit (90- 93) and the second index denotes address to a memory units (90-93)

31. A method according to claim 30,

characterised in,

25   that if LAP is not zero then read commands are generated:

        $R00 = 2*LAP-LAP(N$ down to 0),

        $R10 = R00+2^{minimum(STEP,N)}$,

        $R01 = R00$,

        $R11 = R10$,

30   and generating write commands:

        $W00 = 2*LAP-LAP(N$ downto 0),

        $W10 = W00+2^{minimum(STEP,N)}$,

        $W01 = W00$,

AMENDED SHEET (ARTICLE 19)

W11 = W10.

32. A method according to claim 26 and 28,

characterised in,

5   that data is in bit-reversed order and the read addresses are the same as the write addresses:

    R00 = W00 = 2*LAP-LAP(N down to 0),

    R10 = W10 = W00+2$^{minimum(STEP,N)}$,

    R01 = W01 = W00,

    R11 = W11 = W10.

10

33. Method according to claim 25,

characterised in,

that an order for calculation of the butterfly operators is less than 2 data reads/writes from/to 2

port memory units at each calculation step.

15
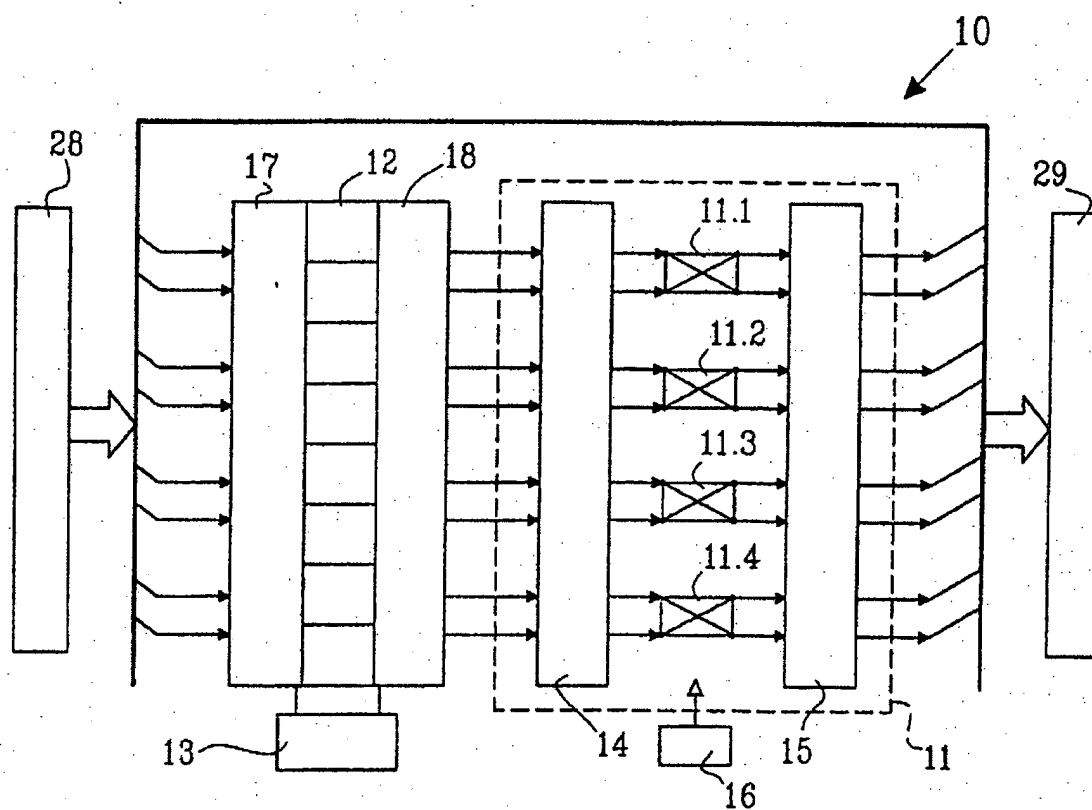
FIG.1

FIG.2
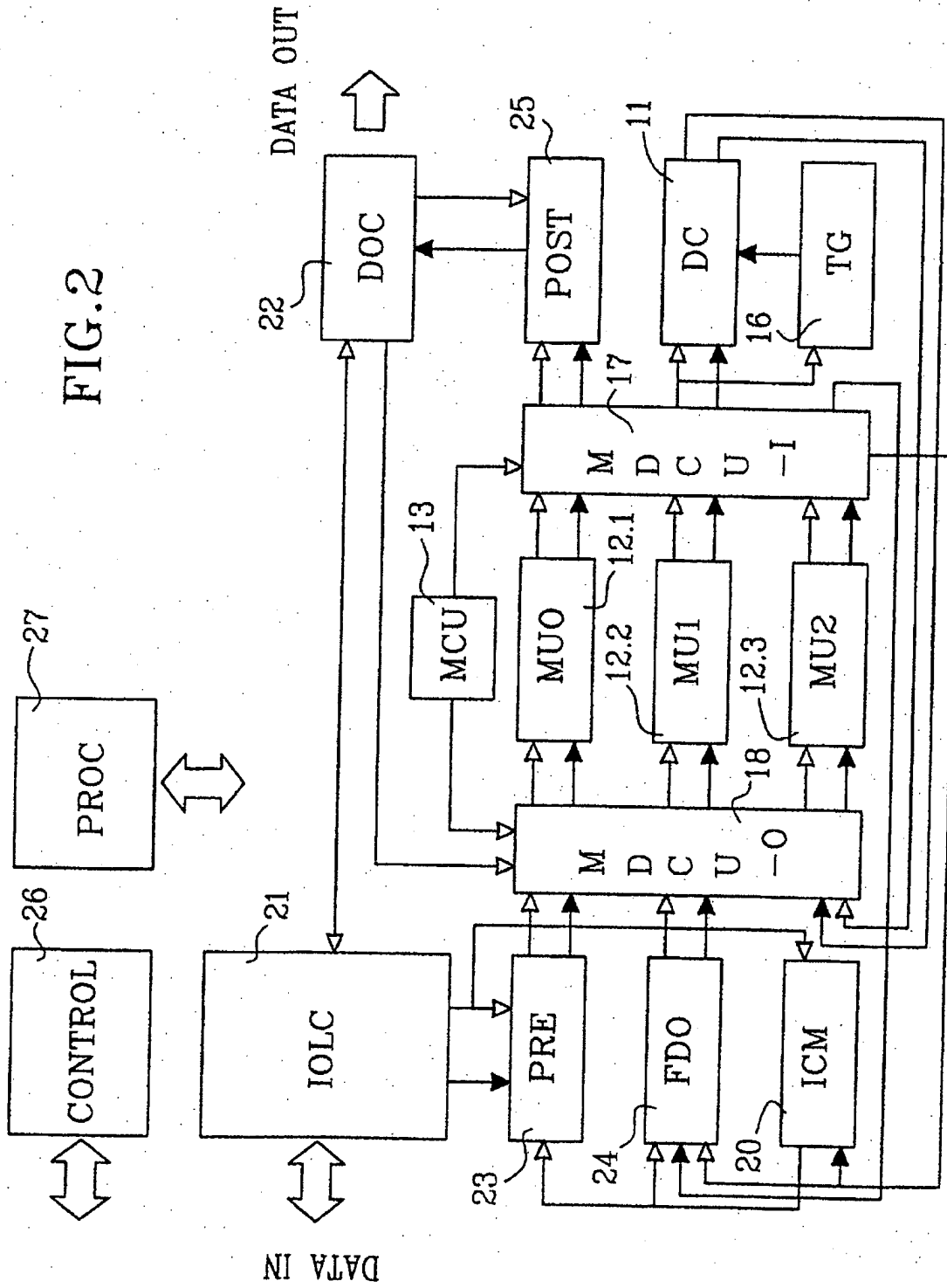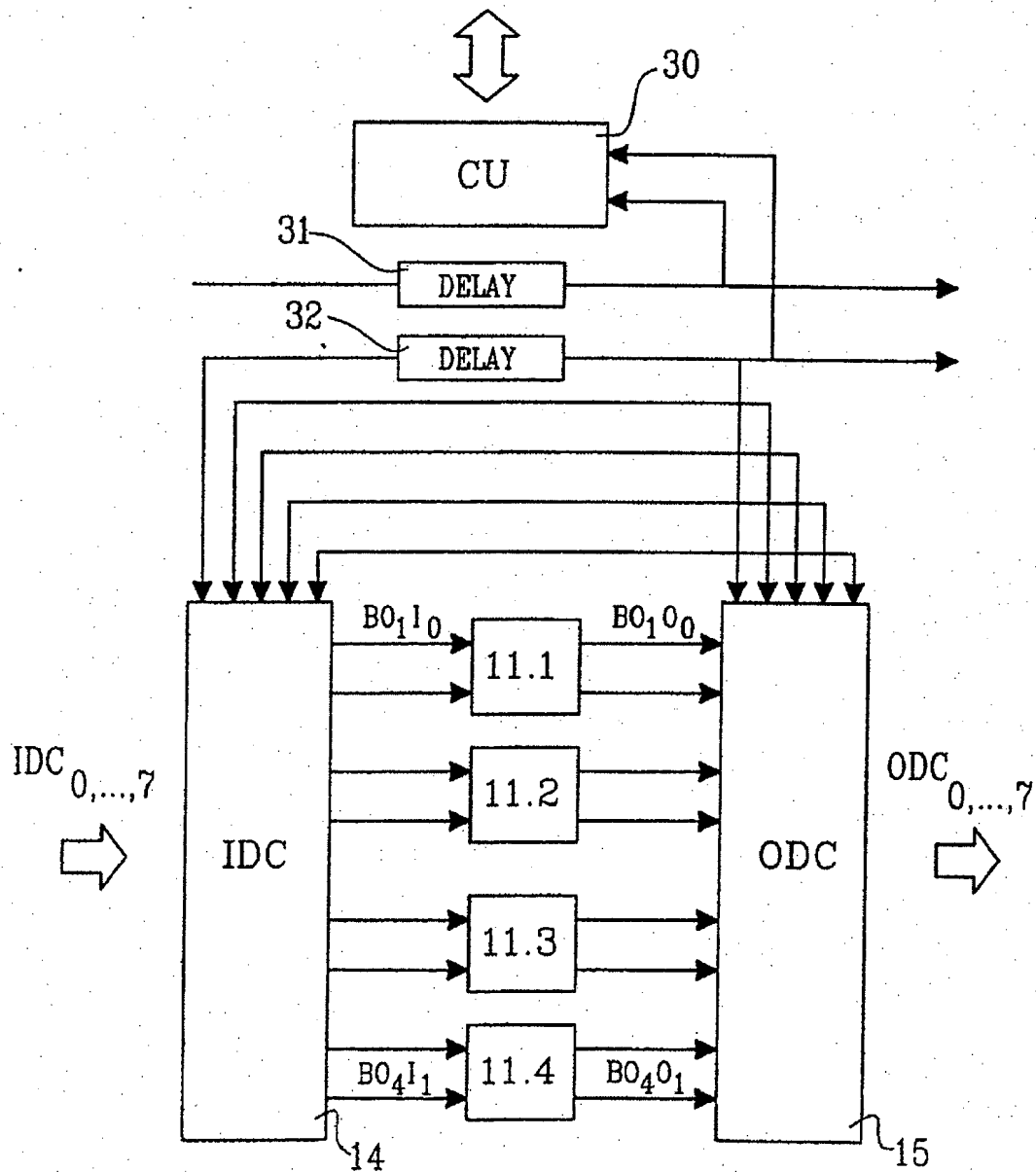
FIG.3
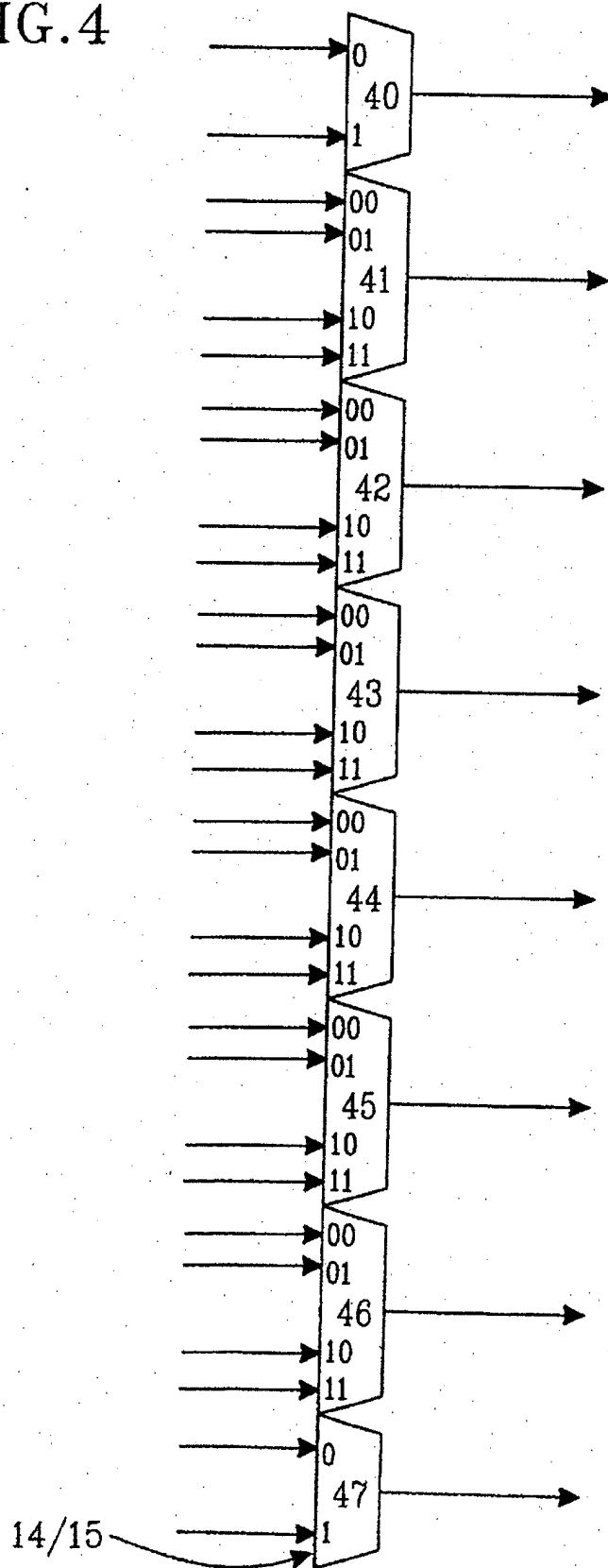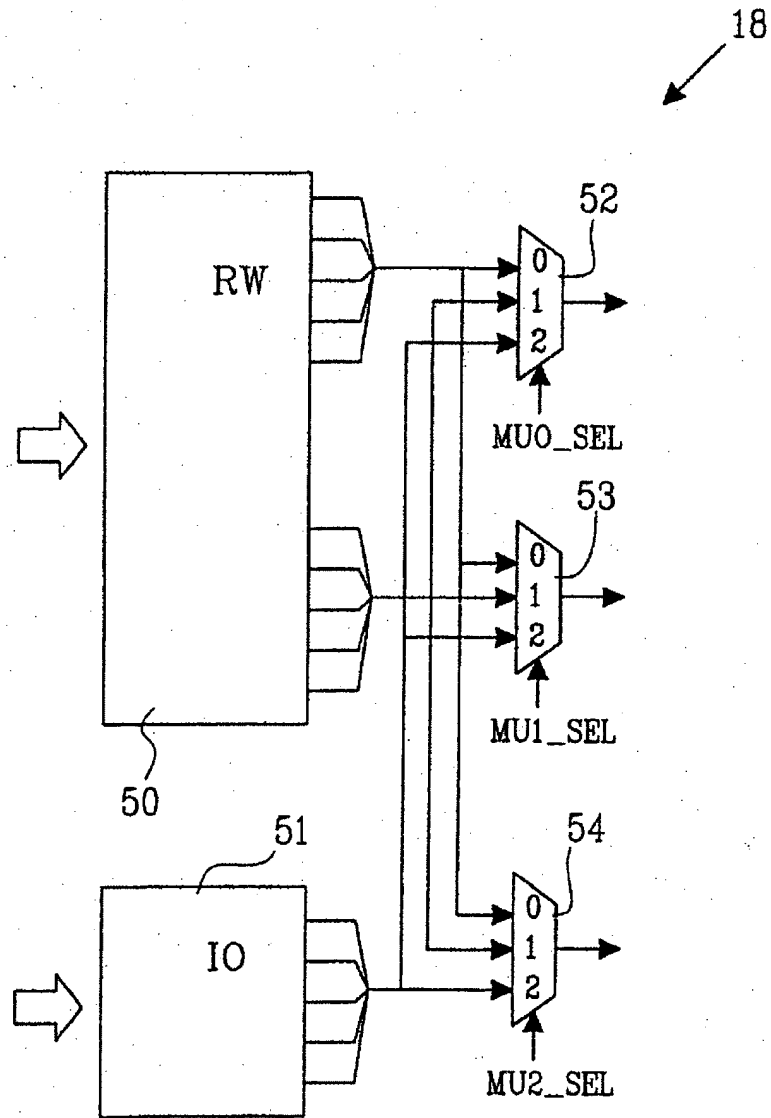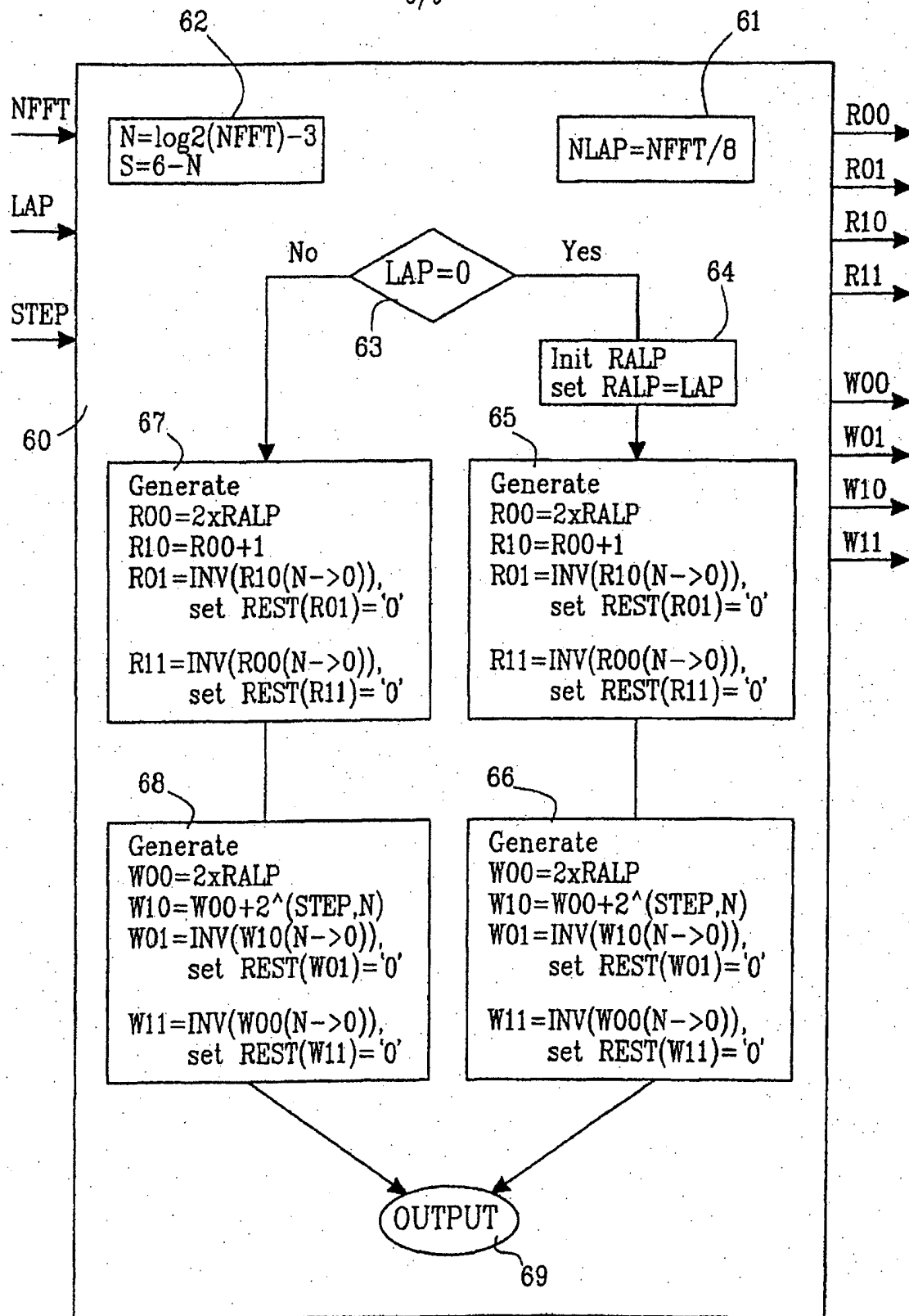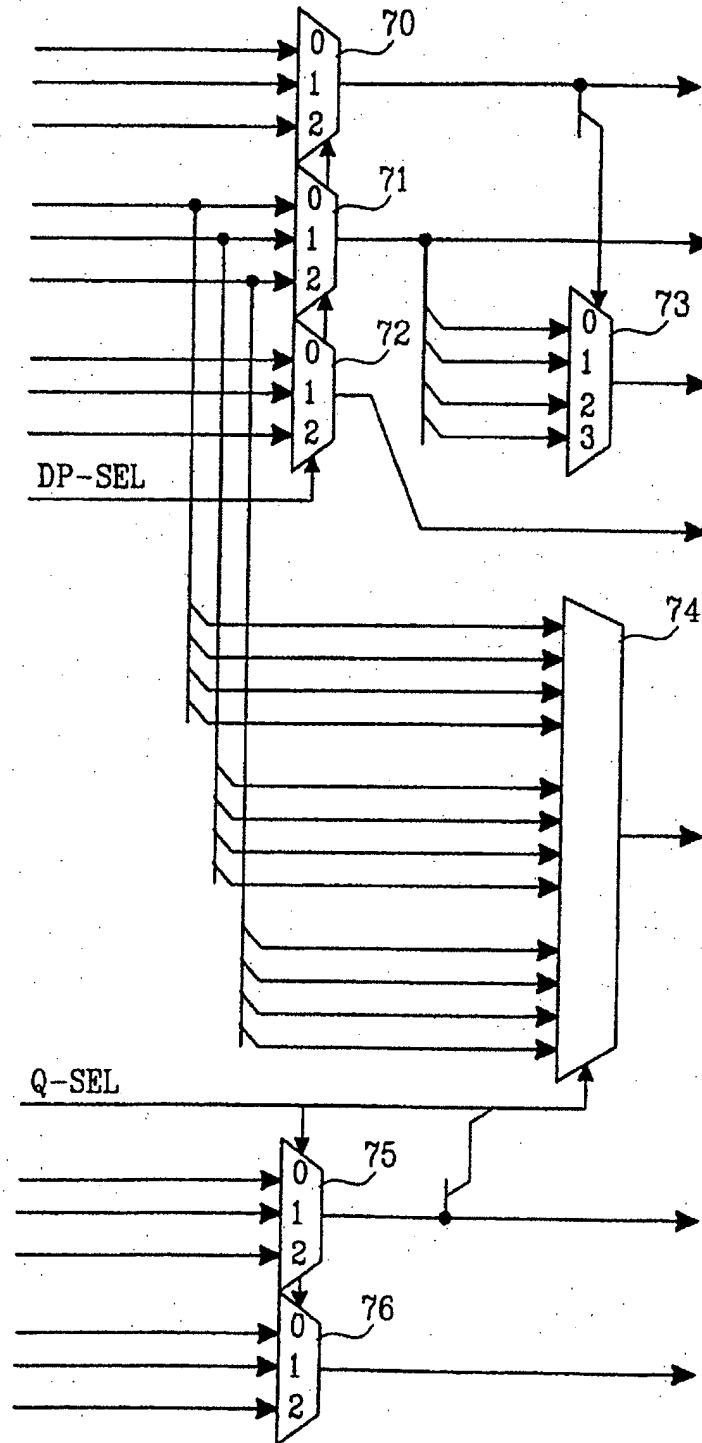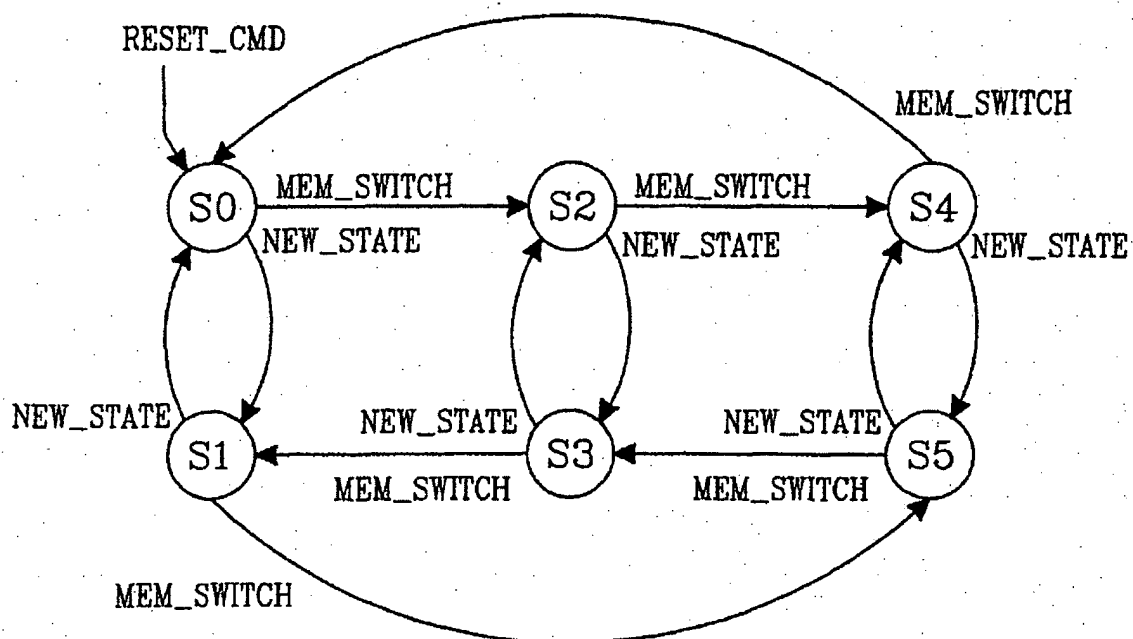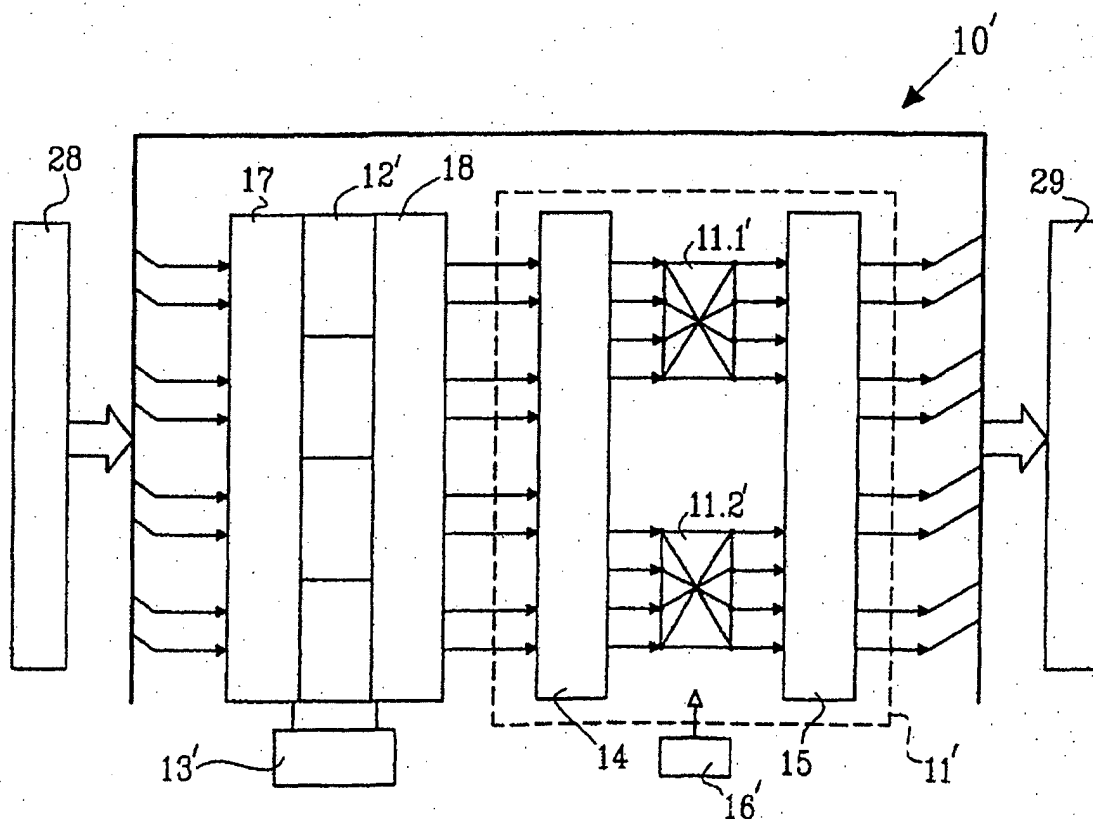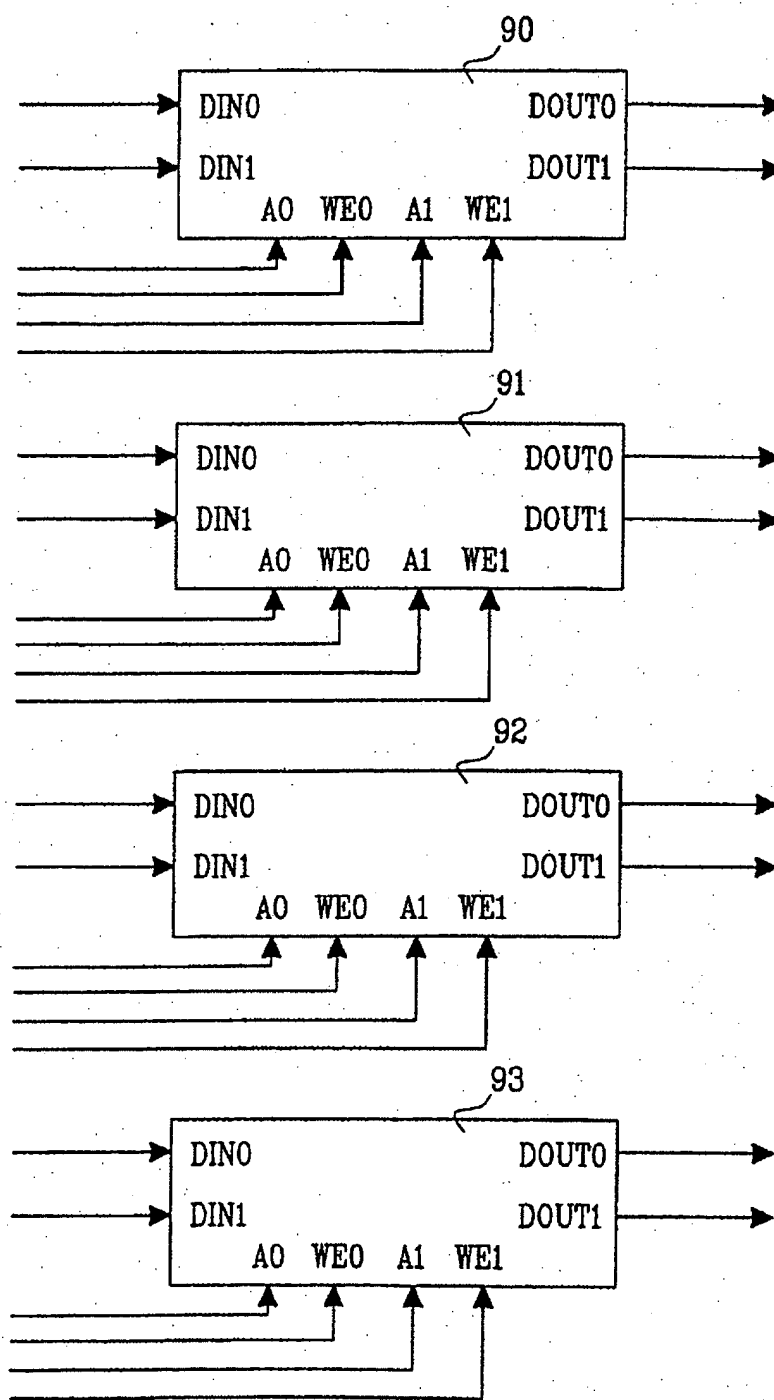
FIG.4

18

FIG.5

6/9

FIG.6

FIG.7

FIG.8

FIG.10

FIG.9

INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER

IPC6: G06F 17/14

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC6: G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 4241411 A (NORMAN F. KRASNER ET AL), 23 December 1980 (23.12.80), column 2, line 30 - column 4, line 14; column 4, line 50 - line 59; column 5, line 21 - line 30, figure 7, column 7, line 23 - line 39	1
Y		3-6,11-18, 20,22-29
A		2
	--	
Y	US 4393457 A (BERNARD J. NEW), 12 July 1983 (12.07.83), column 2, line 18 - line 32; column 4, line 49 - column 5, line 23, figure 2	3-6,11-18, 20,22-29
	--	

[X] Further documents are listed in the continuation of Box C.   [X] See patent family annex.

*	Special categories of cited documents:	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	
"E"	earlier document but published on or after the international filing date	"X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"O"	document referring to an oral disclosure, use, exhibition or other means	
"P"	document published prior to the international filing date but later than the priority date claimed	"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
19 November 1999	2 4 -11- 1999

Name and mailing address of the ISA/	Authorized officer
Swedish Patent Office Box 5055, S-102 42 STOCKHOLM Facsimile No. + 46 8 666 02 86	Erik Veillas/cs Telephone No. + 46 8 782 25 00

INTERNATIONAL SEARCH REPORT

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0805401 A1 (SONY CORPORATION), 5 November 1997 (05.11.97), column 4, line 37 - column 5, line 1; column 7, line 1 - column 8, line 34; column 19, line 27 - column 20, line 9, figure 5	1-35

Patent document cited in search report			Publication date	Patent family member(s)			Publication date
US	4241411	A	23/12/80	NONE			
US	4393457	A	12/07/83	DE EP JP JP WO	3279091 0074401 2030540 58500425 8203483	A A,B B T A	10/11/88 23/03/83 06/07/90 17/03/83 14/10/82
EP	0805401	A1	05/11/97	JP US JP	9297753 5890098 9305573	A A A	18/11/97 30/03/99 28/11/97